www.roavs.com EISSN: 2223-0343

RESEARCH OPINIONS IN ANIMAL & VETERINARY SCIENCES

Research Article

Analyses and comparison of K-nearest neighbour and AdaBoost algorithms for genotype imputation

Abbas Mikhchi[1], Mahmood Honarvar[2], Nasser Emam Jomeh Kashan[1*], Saeed Zerehdaran[3]and Mehdi Aminafshar[1]

[1]Department of Animal Science, Science and Research Branch, Islamic Azad University, Tehran, Iran
[2]Department of Animal Science, Shahr-e-Qods Branch, Islamic Azad University, Tehran, Iran
[3]Department of Animal Science, Ferdowsi University of Mashhad, Mashhad, Iran

Article history
Received: 28 May, 2015
Revised: 21 Jul, 2015
Accepted: 22 Jul, 2015

Abstract

Genomic selection has become a standard tool in dairy cattle breeding. However, for other animal species, implementation of this technology is hindered by the high cost of genotyping. Genotypic imputation is defined as the prediction of genotypes for both unrelated individuals and parent-offspring trios at the single nucleotide polymorphism (SNP) locations in a sample of individuals for which assays are not directly available. Several imputation methods are available for imputation designed for livestock population. Machine learning methods have been used in genetic studies to build models capable of predicting missing values of a marker. In this study, strategies and factors affecting the imputation accuracy of parent-offspring trios were compared using two Machine Learning methods namely K-Nearest neighbour (KNN) and AdaBoost (AB). The methods employed using simulated data to impute the un-typed SNPs in parent-offspring trios. Two datasets of D1 (100 trios with 5k SNPs) and D2 (500 trios with 5k SNPs) were simulated. The methods were compared in terms of imputation accuracy and computation time and factors affecting imputation accuracy (sample size). Comparison of two methods for imputation showed that the KNN outperformed AB for imputation accuracy. The time of computation was different between methods. The KNN was the fastest algorithm. Accuracy of imputation increased with increasing number of trios. Simulation datasets showed that our methods performed very well for imputation of un-typed SNPs and can be used as an alternative for imputation of parent-offspring trios than other methods.

Keywords: Trios; machine learning methods; imputation accuracy; computation time

To cite this article: Mikhchi A, M Honarvar, N Emam Jomeh Kashan, S Zerehdaran and M Aminafshar, 2015. Analyses and comparison of k-nearest neighbour and AdaBoost algorithms for genotype imputation. Res. Opin. Anim. Vet. Sci., 5(7): 295-299.

Introduction

Genomic selection (GS) is a relatively new breeding methodology (Hayes et al., 2009; Lorenz et al., 2011) which is increasingly attractive for the genetic improvement of various species because of its potential to increase the rate of genetic gain (Rutkoski et al., 2013). Genomic selection refers to the use of large numbers of single nucleotide polymorphisms (SNPs) spread across the genome for breeding value estimation and subsequent selection of individuals based on gnomically enhanced breeding values. This

***Corresponding author:** Nasser Emam Jomeh Kashan, Department of Animal Science, Science and Research Branch, Islamic Azad University, Tehran, Iran; E-mail: Nasser_ejk@yahoo.com

technique has become a standard tool in dairy cattle breeding schemes, where it shortens the generation interval substantially (Wellmann et al., 2013). A number of SNP chips from Illumina (http://www.illumina.com) and Affymetrix (http://www.affymetrix.com) are available for cattle. In addition next generation sequencing technologies for low-cost sequencing of whole genomes are now available. Also a major challenge in implementing genomic selection in most species is the cost of genotyping (Khatkar et al., 2012). The prediction of genetic values using models for GS usually requires a first step in which missing genotypes are imputed. Genotype imputation is an important process of predicting unknown genotypes, which uses reference population with dense genotypes to predict missing genotypes for both human and animal genetic variations at a low cost (Boichard et al., 2012; Chen et al., 2013). Imputation methods used to infer missing or un-typed SNP genotypes based on known information (e.g. linkage disequilibrium between missing or un-typed SNPs and their flanking typed SNPs) can provide partial solutions for recovering missing or un-typed genotype data (Pei et al., 2008). Imputation methods can be divided into family-based methods (which use linkage information from close relatives) and population-based methods, which use population linkage disequilibrium information (Sargolzaei et al., 2014). A “trio” data consist of genotypes from father-mother-child triplets and some phasing algorithms are adapted to be used in this type of data (Lu et al., 2014). The accuracy of imputation depends on several factors, such as the number of SNPs in the low density panel, the relationship between the animals genotyped, the effective population size, and the method used (Duarte et al., 2013). The performance of different imputation programs depends mostly on the data structure, e.g., density of single nucleotide polymorphism (SNP) panels, size of the reference population, and whether related or unrelated individuals were genotyped. Thus, choosing the best imputation method for a given data set is not straightforward (Wellmann et al., 2013). Machine learning methods have been used in genetic studies to explore the underlying genetic profile of disease and build models capable of (i) detecting gene-gene interactions; (ii) predicting disease susceptibility; (iii) predicting cancer recurrence; and (iv) predicting missing values of a marker (Wang et al., 2012; Goddard et al., 2013). The K-Nearest neighbour (KNN) algorithm belongs to the category of instance-based learners which is simple and an important machine learning algorithms (Sun and Zhao, 2015). Boosting is an ensemble based method which attempts to boost the accuracy of any given learning algorithm by applying it several times on slightly modified training data and then combining the results in a suitable manner. In this research the accuracies of AdaBoost and K-Nearest neighbor (KNN) algorithms for imputation of un-typed-SNPs of parent-offspring trios are compared. The methods were compared in terms of imputation accuracy, computation time and factors affecting imputation accuracy. To evaluate the factors affecting imputation accuracy, sample size and SNP density were also examined.

Materials and Methods

Data simulation

Data sets with 5 chromosomes each had 100 cM were simulated to allow comparison of AdaBoost and K-Nearest neighbour (KNN) algorithms, in term of accuracy of imputation. An effective population size of 100 animals was simulated, of which half of the animals were female and the other half male. This structure was kept constant for 50 generations using mutation rate of 2.5 x 10^{-8} per site by drawing the parents of an animal randomly from the animals of the previous generation. Mating was performed by drawing the parents of an animal randomly from the animals of the previous generation. The number of markers was 5000 for each chromosome. Finally, the population consisted of 100 and 500 trios and each trios contained dam, sire and offspring. For each of D1 (100trios) and D2 (500 trios) datasets five versions (NA10, NA30, NA50, NA70 and NA 90) were created with different levels of simulated missing data (10, 30, 50, 70 and 90 percent of offspring genotypes). All the imputations in this study were done using MATLAB software version (R2014a).

Imputation accuracy and running time

For each method, the imputation accuracy per un-typed SNPs in offspring were calculated as the correlation between imputed and observed SNPs, then mean of imputation accuracy were calculated across the 5 replicates. Computation time were measured based on running each program in second on a windows server with 32 core CPU Intel, GPU: 192 CUDA Core and a total of 64 GB RAM by profiler function in MATLAB.

Assessment of factors affecting imputation accuracy

The sample size was considered as factor that could impact the imputation accuracy. For each dataset-imputation method combination, imputation accuracy was averaged across dataset versions NA10, NA30, NA50, NA70 and NA90 and referred as imputation accuracy. To assess the effect of the sample size on imputation accuracy, two groups of 100 and 500 parent-offspring trios were created. For both groups embedded simulated SNPs with 5k panel and compared imputation accuracy based on trios sample size. The impact of each of these factors was assessed for each imputation method.

Imputation methods

AdaBoost

The AdaBoost algorithm is a well-known method to build ensembles of classifiers with very good performance ((Sateesh et al., 2012). It has been shown empirically that AdaBoost with decision trees has excellent performance, being considered the best off-the-shelf classification algorithm (Sateesh et al., 2012). This algorithm takes training data and defines weak classifier functions for each sample of training data. Classifier function takes the sample as argument and produces value 0 or 1 in case of a binary classification task and a constant value-weight factor for each classifier. Generally, AdaBoost has shown good performance at classification. The sensitivity to noisy data and outliers is a weak feature of AdaBoost. Let X be a set of imputed SNPs, and y be a vector of observed ('true') SNP at an individual. Define M=100 to be the number of independent classifiers (i.e. the imputation software). Given a training set of N SNP, there are $Z=[(x_1, y_1), \dots ,(x_i, y_i), \dots ,(x_N, y_N)]$, where $x^i \in X=(x_{i1}, x_{i2}, x_{i3}|i=1,2, \dots , N)$, $y^i \in y=(a_1, a_2)$, and a_1, a_2 are the two alleles at a SNP locus, in question, for SNP i in the training sample.

Initialize: each SNP was assigned with an equal weight

$$w_i = \frac{1}{N}, i \in \{1, \dots, N\}$$

Training: For m=1, 2… M classifiers

Call classifier m, which in turn generates hypothesis P_W (i.e. inferred SNPs in the training set). Calculate the error of P_W:

Fit the class probability estimate

$P_m(x) = P_w(y = 1|x) \in [0 , 1]$, using weight w_i on the training data.

Set $H_m = 0.5 \log\left(\frac{1 - P_m(x)}{P_m(x)}\right) \in R$

Update the weight distribution W_i for next classifier as Set $w_i \leftarrow w_i \exp(-w_i H_m(x_i))$ and renormalize to

$$\sum_i w_i = 1$$

Testing: In the testing set, each Un-typed SNP is classified via the so-called 'weighted majority voting'. Briefly, the wrapper program is

Output H(x) = sign $\left(\sum_m^m H_m(x)\right)$

Above, the algorithm maintains a weighted distribution W_i of training samples x_i, for i=1, …, N, from which a sequence of training data subsets Z_m is chosen for each consecutive classifier (package) m. Initially, the distribution of weights is uniform, meaning that all samples contribute equally to the error rate. Next, the logit H_m of the rate of correctly classified samples is calculated for classifier m. A higher H_m is an indicator of better performance. For instance, when H_m=0.5, H_m takes the value 0, and increases as $H_m \rightarrow$ 0.

K-Nearest Neighbor (KNN)

The K- Nearest Neighbors is a simple algorithm that stores all available cases and classifies new cases based on a similarity measure. There are two major design choices to make the value of k, and the distance function to use. The KNN classifier classifies input data according to the labels of the K-Nearest Neighbours in the training data $\{\mathbf{x_i}, y_i\}$. The Euclidean distance between a training sample and the target sample calculated and both represented as *(L + R)*-dimensional binary vectors (input data point $\mathbf{x_u}$, and each of the training data point's $\mathbf{x_i}$) using:

$$d\ (X_u, X_i) = \|X_u - X_i\|$$

The category label of $\mathbf{x_u}$ was assigned based on the majority vote of the category labels of its k-nearest training data (KNN) (Sun and Zhao, 2015):

$$\delta(X_u) = argmax \sum_{x_i \in kNN} \delta(X_i, Y_j)$$

Where $\delta(\mathbf{x_i}, y_j) \in \{0, 1\}$ indicates if $\mathbf{x_i}$ belongs to y_i. We tested k = 3 and used the fitknn function implemented in MATLAB.

Results

The imputation accuracies of two datasets are shown in Table 1 for KNN and AB. The accuracy of imputation was high for two imputation methods. For all data sets, imputation accuracies always decreased as the level of missing data increased. In general AB had the lowest imputation accuracy compared to KNN methods, however, the difference between imputation methods was not significant. The accuracy of imputation increases with increasing the sample size for all imputation methods examined here. The imputation accuracy was lower for all levels of 100 compared to

Table 1: Mean of imputation accuracy for each imputation method in various versions on the two datasets

AB	KNN	Version	Sample size	Density	Data set
0.9843	0.9945	NA10	100	5k	D1
0.9883	0.9958	NA30	100	5k	
0.9822	0.9932	NA50	100	5k	
0.9777	0.9901	NA70	100	5k	
0.9211	0.9373	NA90	100	5k	
0.9707	0.9821	Mean			
0.9859	0.9969	NA10	500	5k	D2
0.9885	0.9958	NA30	500	5k	
0.9877	0.9940	NA50	500	5k	
0.9800	0.9903	NA70	500	5k	
0.9288	0.9739	NA90	500	5k	
0.9741	0.9901	Mean			

KNN= K-Nearest Neighbours, ADA= AdaBoost, NA10: 10% of genotype is missing per offspring, NA30: 30% of genotype is missing per offspring, NA50: 50% of genotype is missing per offspring, NA70: 70% of genotype is missing per offspring, NA90: 90% of genotype is missing per offspring

Table 2: Average imputation runtime on six datasets in seconds

Data Set	Sample size	Density	Version	KNN	AB
D1	100	5K	NA90	45s	2930s
D2	500	5K	NA90	185s	3460s

KNN= K-Nearest Neighbors, ADA= AdaBoost, NA10: 10% of genotype is missing per offspring, NA30: 30% of genotype is missing per offspring, NA50: 50% of genotype is missing per offspring, NA70: 70% of genotype is missing per offspring, NA90: 90% of genotype is missing per offspring

500. The detailed runtime of the two methods on two datasets at missing rate of 90% (NA90) is presented in Table 2. For all data sets, the KNN was always fastest algorithm. AB needed more time to impute a dataset. The AB was always the slowest. An important factor in evaluating machine learning algorithms is how quickly their runtime increases with sample size of dataset. As sample sizes of trios grow, the speed of all t methods needed some more time to impute a dataset.

Discussion

This study applied and compared the performance of two machine learning methods for imputation in parent-offspring trios. Using simulated data sets, it was determined that factors such as number of trios have varying effects on imputation accuracy rates. Abilities of the two methods were compared in terms of imputation accuracy and running time. In comparing running time, KNN had better performance than AB algorithm. For KNN Euclidean distance function and k was a fixed number across all variables. The imputation accuracy increased but greater computational load of the AB method in compare with KNN is due to its adaptive weighting of variables that takes into account the response variable (Rutkoski et al., 2013). In general, AdaBoost Algorithm had lower imputation accuracy than KNN methods. A possible reason that AdaBoost were less accurate than KNN method is that the datasets that we used in our experiment may have violated multivariate normality. In addition, increasing the total number of trees can improve boosting ability to impute the un-typed SNP. Nevertheless other reason that affective on decreasing the accuracy may be due to total number of trees that we used in our experiment. KNN required less computer time than Adaboost, which may be an advantage when using large data sets with several thousand markers. The results showed that KNN had better performance than AB when we used a 5k SNP panel with a missing rate 90%. This can be due to Euclidean distance function in KNN algorithm. Comparing imputation accuracy obtained by our methods with other studies is hard because each study uses different population structure, levels of missing data, and levels of LD between markers (Rutkoski et al., 2013). As a general trend, accuracy of imputation increased with increasing number of trios. It seems that imputation accuracy in all methods was influenced by sample size. Larger sample size will produce more consistent estimates of measured parameters resulting in generally improved imputation accuracy for various methods (Okser et al., 2014). The performance of any classification depends on sample size, which may be especially so for our methods, since the number of parameters to be estimated is large and low sample size may lead to unstable results (Sun and Zhao, 2015). On the other hand, computing time changed with increasing the sample sizes. For all methods, when sample size increased from 100 to 500, the computing time also increased.

Conclusion

This study applied and compared the performance of two machine learning methods for imputation in parent-offspring trios. This study has important implications when all parents has been genotyped completely, a trio-based algorithm can be used to impute the high density genotypes of offspring genotyped with a lower density panel. With the algorithms presented here, the imputation of un-typed SNP was more accurate. Simulation datasets showed that our methods performed very well for imputation of un-typed SNPs and can be as an alternative for imputation of parent-offspring trios than other methods.

Acknowledgements

We would like to express our gratitude to all those who helped to complete this paper, especially for Dr Y Forghani, Dr M Kamaei, Dr Y Bernal Rubio, constructive suggestions and encouragements help us in all the time of this research.

References

Boichard D, Chung H, Dassonneville R, David X, Eggen A, Fritz S, Gietzen KJ et al. (2012) Design of a bovine low-density SNP array optimized for imputation. PLoS ONE 7(3): e34130.

Chen J, Zhang JG, Li J, Pei YF, Deng HW (2013) On Combining reference data to improve imputation accuracy. PLoS ONE 8: e55600.

Duarte JLG, Bates RO, Ernst CW, Raney NE, Cantet JC, Steibel JP (2013) Genotype imputation accuracy in a F2 pig population using high density and low density SNP panels. BMC Genet 14: 38.

Goddard R, Eccles D, Ennis S, Rafiq S, Tapper W (2013) Support vector machine classifier for estrogen receptor positive and negative early-onset breast cancer. PLoS ONE 8: e68606.

Hayes BJ, Bowman PJ, Chamberlain AJ, Goddard ME (2009) Genomic selection in dairy cattle: progress and challenges J Dairy Sci 92: 433–443.

Khatkar MS, Moser G, Hayes BJ, Raadsma HW (2012) Strategies and utility of imputed SNP genotypes for genomic analysis in dairy cattle. BMC Genomics 13: 1471-2164.

Lorenz AJ, Chao S, Asoro FG, Heffner EL, Hayashi T (2011) Genomic selection in plant breeding: knowledge and prospects. Adv Agron, 110: 77-123.

Lu AT, Cantor RM (2014) Identifying rare-variant associations in parent-child trios using a Gaussian support vector machine. BMC Proceedings 8(Suppl 1): S98.

MATLAB (2014) http://www.mathworks.com

Okser S, Pahikkala T, Airola A, Salakoski T, Ripatti S, Aittokallio T (2014) Regularized Machine Learning in the Genetic Prediction of Complex Traits. PLoS Genet 10: e1004754.

Pei YF, Li J, Zhang L, Papasian CJ, Deng HW (2008) Analyses and comparison of accuracy of different genotype imputation methods. PLoS ONE 3: e3551.

Rutkoski JE, Poland J, Jannink J, Sorrells ME (2013) Imputation of unordered markers and the impact on genomic selection accuracy. G3 (Bethesda) 3: 427–439.

Sargolzaei M, Jansen GB, Schenkel FS (2014) A new approach for efficient genotype imputation using information from relatives. BMC Genomics 15:478.

Kumar BS (2012) Boosting techniques on rarity mining. Int J Adv Res Comput Sci Softw Eng 2: 27-35.

Sun J, Zhao H (2015) The application of sparse estimation of covariance matrix to quadratic discriminant analysis. BMC Bioinformatics 16:48

Wang Y, Cai Z, Stothard P, Moore S, Goebel R, Wang L, Lin G (2012) Fast accurate missing SNP genotype local imputation. BMC Res Notes 5: 404.

Wellmann R, Preub S, Ernst E, Heinkel J, Wimmers K and Bennewitz J (2013) Genomic selection using low density marker panels with application to a sire line in pigs. Genet Select Evol 45: 28.